COMSC-210 Lab 3
A Static Indexed Array Template

GOOD PROGRAMMING PRACTICES *show / hide*

ABOUT THIS ASSIGNMENT
In this lab session, you will write and test your first full data structure template.

After you complete this lab assignment, post the required files to the COMSC server so that you receive the proper credit for this 50-point lab. Your posted files are due at midnight of the evening of the due date indicated in the course outline. Use the "Submit your FA2015 work" link on the class website to post your file for this lab, using the **lab3** folder provided.

LAB 3a: Write A Static Array Class Template [`StaticArray.h and StaticArrayDriver.cpp`]
Purpose. The purpose of this lab is for you to write a templated class. The resulting template can be used in any program in place of a C++ array, in case you want the advantages of range safety and built-in size tracking.

Requirements. Write **`StaticArray.h`** per our lecture topic 3 notes, and **`StaticArrayDriver.cpp`** to test it. Name the template `StaticArray`, with the data type and the capacity as template variables (as in the lecture topic notes). Write the public interface exactly as specified in the lecture topic notes -- do not add to, or change the public interface as specified.

Write a test driver named **StaticArrayDriver.cpp**, testing all functions, ifndef, object copy, and object assignment. Make sure you test the square bracket getter -- you'll need a read-only copy of a StaticArray object for this:

```
StaticArray<int, 10> a;
...
const StaticArray<int, 10> copy = a;
for (int i = 0; i < copy.capacity(); i++)
  cout << "copy[" << i << "] = " << copy[i] << endl;
```

NOTE: If you include a data member to track capacity, you are not doing this right!

DO NOT depend on other included files to include the library item(s) that your H or CPP need.
DO #include in each H and CPP the library items used in that H or CPP.
DO write "using namespace std;" below the C++ library includes in any H or CPP that has them.

Submit the two files (1 CPP and 1 H) to the class website for credit.

LAB 3b: Write A Static Array Application [`MyStaticArray.cpp`]
Purpose. The purpose of this lab is for you to apply your own templated class. If it was fully tested in 3a above, then it should work fine in this application.

Write **`MyStaticArray.cpp`**, using your **StaticArray.h** template. Write the app for a data type of your choosing (int or double), sized for a maximum of 100 values. The app lets a user enter as many values as they like (up to 100), and when that process is completed, lets the user look up values by matching key.

The app should start by prompting the user to enter a pair of numbers on the same line: an index and its corresponding value. Use the square bracket operator to store the value at the index. Do *not* validate input in the app, because your template should handle out-of-range indexes, and it should allow overwriting at an "in use" index. Quit the loop when an uppercase or lowercase Q is entered for either the index or the value.

After all data entry is complete, the app should:

1. output the data structure *size* after Q was entered,
2. output the list of all entered values with their indexes, per the example below, and
3. implement an event-controlled loop that prompts for an index value and outputs whether the index is in use or not, and if in use, what is the value stored for that index. Allowing multiple searches until the user elects to stop by entering uppercase or lowercase Q.

Here's a sample of how this should work (user input in blue):

```
Input an index and a value [Q to quit]: 33 12
Input an index and a value [Q to quit]: 4 100
Input an index and a value [Q to quit]: 5 300
Input an index and a value [Q to quit]: x 17
Input an index and a value [Q to quit]: 33 120
Input an index and a value [Q to quit]: -1 234
Input an index and a value [Q to quit]: q

I stored this many values: 4
The values are:
  0 17
  4 100
  5 300
  33 120

Input an index for me to look up [Q to quit]: 33
Found it -- the value stored at 33 is 120
Input an index for me to look up [Q to quit]: 1000
I didn't find it
Input an index for me to look up [Q to quit]: Q
```

Design the prompts and the output as you like.

DO NOT do any testing in this CPP -- it's an "app". That means no ifndef testing!

Remember to NOT read numerics directly from `cin` -- read them into a string buffer and convert.

How to pause a console program: *show / hide*

GRADING RUBRIC *show/hide*